

代码说明

一、需求背景

本代码系分析课表管理的一部分功能。

旨在以面向对象的方式实现课表的状态展示，查看等功能。课表数量繁多，而每个课表的功能类似，为了提高代码的复用性，减少代码冗余，以及使得功能能够模块化。进而产生了使用 javascript 类及面向对象的概念来进行设计的思路。

二、设计思路

1. 分析课表模块的各项子功能。
2. 提取该模块中的实例属性和公用方法。
3. 使用构造函数初始化私有变量，使用原型链实现公用方法的继承。
4. 组合构造函数及原型链两种方式来模拟创建类

三、实现说明

1. 功能拆分



课表模块

时间刻度
高亮

点击向左
滚动

点击向右
滚动

悬浮滚动

按钮状态
检测

2. 实例属性和公用方法提取

2.1 实例属性：

- 课程元素对象
- 课程时间刻度元素对象
- 左点击按钮元素对象
- 右点击按钮元素对象
- 课程总页数
- 当前页数

2.2 公用方法

- 课程向左滚动
- 课程向右滚动
- 滚动过程时时控制按钮状态
- 滚动过程课程时长刻度高亮状态同步变化
- 课程时长刻度悬浮，显示对应课程

3. 采用构造函数方式初始化所有实例属性

```
9
10 //课程展示 (课表)
11 function CoursePlay(course,n){
12     //下面是设置coursePlay的属性
13     this.times = 0; //初始化计数器
14     this.course = $(course).eq(n); //选到ul
15     this.courseInfo = this.course.find(".courseInfo"); //选到ul的li
16     this.hasClass = this.course.closest(".courseList").siblings('.timeInfo').find(".has-class"); //取到刻度上有课程的块
17     this.multiple = Math.ceil(this.courseInfo.length / 4); //对4取余向上取舍 (假如有10个li 10/4=3)
18     this.leftMove = this.course.parent().siblings(".leftMove"); //选取左边按钮
19     this.rightMove = this.course.parent().siblings(".rightMove"); //选取右边按钮
20     this.titleActive();
21     //初始化左右按钮的状态 (可点/不可点)
22     this.checkButtonStatus();
23     var _this = this;
24     //点击左移
25     this.leftMove.click(function(){
26         _this.moveToLeft();
27     });
28     //点击右移
29     this.rightMove.click(function () {
30         _this.moveToRight();
31     });
32 }
```

4. 采用原型链方式实现公用方法的继承

```
//左移
CoursePlay.prototype.moveToLeft = function(){
    //假设li的长度为10, 则 this.multiple=3
    var a = ++this.times; //times初始是0, 左点击一次, times+1, 则a=1
    var courseWidth = this.course.width(this.course.find(".courseInfo").length * 283.7);
    if (a < this.multiple) { //1<3
        this.course.stop().animate({"left": -283.7 * 4 * a + "px"}, 1000); //向左平移4个li
    } else {
        this.times = this.multiple-1; //将times设置为3-1=2
    }
    this.titleActive();
    this.checkButtonStatus();
};

//右移
CoursePlay.prototype.moveToRight = function(){
    var b = --this.times;
    var courseWidth = this.course.width(this.course.find(".courseInfo").length * 283.7);
    if (b >= 0) {
        this.course.stop().animate({"left": -283.7 * 4 * b + "px"}, 1000);
    } else {
        this.times = 0;
    }
    this.titleActive();
    this.checkButtonStatus();
};

CoursePlay.prototype.titleActive = function(){
    this.hasClass.removeClass('selected');
    var pageStart = this.times*4;
    var pageEnd = (this.times+1)*4;
    for(var i=pageStart; i<pageEnd; i++){
        this.hasClass.eq(i).addClass('selected');
    }
};

//左右按钮状态
CoursePlay.prototype.checkButtonStatus = function(){
    if(this.times <= 0){ //右按钮的显示状态
        this.rightMove.css({"cursor":"not-allowed","color": "#eeeeee"});
    }else{
        this.rightMove.css({"cursor":"pointer","color": "#d2d2d2"});
    }
    if(this.times >= this.multiple-1){ //左按钮的显示状态
        this.leftMove.css({"cursor":"not-allowed","color": "#eeeeee"});
    }else{
        this.leftMove.css({"cursor":"pointer","color": "#d2d2d2"});
    }
};
```

四、代码文件

1. manageClass.html 页面结构
2. manageClass.css 页面布局
3. manageClass.js 课程展示功能模块

五、总结

1. 面向对象编程，独立开发各个子功能，实现代码复用。
2. 结合构造函数及原型链的方式模拟类，避免了构造函数重复声明方法（占内存）的缺陷。